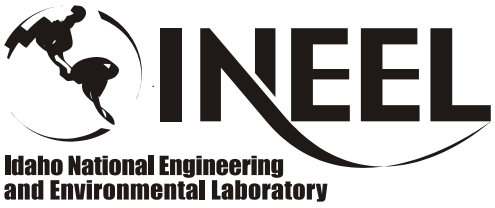
INEEL/CON-02-00584
PREPRINT



Maximizing Return From Sound Analysis and Design Practices

J. D. Bramlette

June 15, 2002 – June 22, 2002

Oracle Development Tools User Group (ODTUG) 2002

MAXIMIZING RETURN FROM SOUND ANALYSIS AND DESIGN PRACTICES

Judy Bramlette, Idaho National Engineering and Environmental Laboratory

Abstract

With today's tightening budgets computer applications must provide "true" long-term benefit to the company. Businesses are spending large portions of their budgets "Re-Engineering" old systems to take advantage of "new" technology. But what they are really getting is simply a new interface implementing the same incomplete or poorly defined requirements as before.

"True" benefit can only be gained if sound analysis and design practices are used. *WHAT* data and processes are required of a system is not the same as *HOW* the system will be implemented within a company. It is the System Analyst's responsibility to understand the difference between these two concepts.

I will be discussing some simple techniques to be used during the Analysis and Design phases of your project. I will discuss the information gathered and recorded in each phase and how it is transformed between these phases. I will also cover choices available and the impact these choices have on downstream phases. All this will be demonstrated using a current production application generated using Oracle Designer[a].

Applying these techniques to "real world" problems, you will find that your applications meet the needs for today's business and adapt easily to your ever-changing business environment.

Project Background

The Idaho National Engineering and Environmental Laboratory (INEEL) has been in operation since 1949. It is a science-based, applied engineering national laboratory dedicated to supporting the U.S. Department of Energy's missions in environment, energy, science and national defense. The INEEL Chemical Management System (ICMS) was developed to provide a tool to help consistently manage and report the chemicals stored and used at this laboratory.

When this project started, there were five separate contracting companies operating various facilities at the INEEL and each had their own method for managing chemicals. This was a problem for two reasons. The first because within less than a year these contracts were being consolidated into a single contract and would be managed by only one company. The second was because for regulatory purposes, the company was required to report as a single unit. Two of the companies had already developed their own automated systems to help in their chemical management effort, but neither met the needs for all the facilities. Some of the facilities at the INEEL use chemicals for research and development, storing and using only small amounts of many different chemicals. While other facilities use chemicals for processing and manufacturing purposes, maintaining and using large amounts of only a few chemicals. It was decided to combine the functionality from these two existing systems and add the new functionality that was missing.

This project began over seven years ago. During this time, the INEEL has gone through 2 major contractor changes resulting in different management philosophies, as well as, organizational changes. The regulations governing what and how chemicals need to be reported are constantly changing and the applications scope has been broadened with the addition of many new enhancements. At the time, the technology industry was moving forward causing changes in the tools used by the development team and the company's acceptable standard hardware/software environment. Because the development team used the techniques discussed in this paper, the application has been able to accommodate these changes without disrupting the core foundation of the application.

The ICMS team started this project using Oracle's CASE 5.1 and has since migrated forward to Designer 2.1.2. Today, the database objects, forms, and reports are generated and maintained through this same repository. The team supported the idea of the repository as a central location to record the information that was gathered. We found the Oracle repository an

[a] PRODUCT DISCLAIMER – References herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government, any agency thereof, or any company affiliated with the Idaho National Engineering and Environmental Laboratory.

excellent tool to use for data modeling. During the generation of the application, it was extremely beneficial to have an automated tool that ensured the data rules were enforced. It is not perfect though. Somewhere to store the business rules about the data collected during analysis would have been beneficial. We ended up having to track these outside of the tool until Design and then most were added as check constraints or triggers to the table definition.

Assumptions

The information in this paper refers to *information* systems only. Techniques needed to develop data acquisition systems and systems that simulate a particular activity are not being addressed.

Italicized terms are defined at the end of this paper. When the word is italicized it is conforming to this exact definition.

Philosophy

One of any companies' most valuable resources is its *information*. It drives most of the daily tasks and decisions made. The quality of information the people have to work with impacts the quality of the job done. And the speed at which this *information* is available, impacts how competitive the business can be in the market place.

Computers are a great place to store *data* and they are generally an efficient way to convert, manipulate, consolidate, and translate *data* into useful *information*. But not every process should be automated. Computer applications can only replicate processes that can be done manually. Manual processes can be anything from storing the data, to summarizing it, to retrieving it for use. By identifying and analyzing *what* is being done, a company will be able to identify the optimal manual process. This is where the greatest inefficiencies are eliminated. From that point forward, the company will only gain "true" benefit if the automated process is faster (and in some cases even feasible), more efficient, and/or more accurate than this optimal manual process. This holds true when "re-engineering" previously automated systems as well. It will also be easier to see which processes would provide the greatest return if automated.

As an Analyst, if you can not identify and understand the manual process which needs to take place, there is no way you will be able automate it. The old saying, "The customer never knows what they want." may be true in most cases. But what they do know is what they do and how they do it. And it is from this information that the Analyst helps the customer discover and understand the requirements they have. This makes the rest of the development process much easier for everyone.

The key to developing applications that "truly" benefit the company, now and in the future, centers on understanding the *data* the company receives and generates, how this *data* relates to other *data*, the business processes that generate, act upon, or use this *data*, and how this *data* is converted into *information* that the customer can use. There are many software life-cycle methodologies published to help the developer achieve and document this understanding. Each methodology goes about it differently, but they still come down to defining this same basic information. However, there are some methodologies, such as prototyping, that try to merge analysis and design together and I feel these are more valuable when treated as a design methodology.

In order to provide long-term benefit to a company, a system must be flexible and allow the customer to easily manage and change the factors that influence the responses from the system, such as changes in policy. The customer should not have to go to the programming staff every time one of these items changes. This is accomplished by understanding what these factors are and basing the systems responses on data that is controlled by the customer versus code written or generated by the developer.

If *what* your business does is not understood or is flawed when design is started, it will cause problems later in the project. Defining the *what* is not a step you can skip. You will do it sooner or later. And doing it first will always be the quickest and definitely the cheapest way to go. If not, simple enhancements or modifications will be harder to do and in some cases even impossible without a re-write, not to mention the more complex ones. The people effort to use the system will be confusing and hinder their job performance. Work-arounds or entirely new processes will be developed to compensate for these flaws. It will also be harder for other developers to learn and understand the system, making the company more dependent on the original developers.

What your business does is a fairly static definition. Changes to this are usually only when a business adds or removes an entire portion of the business process. *How* your business operates and the technology it uses is dynamic. The information gathered during analysis defines *what* your business does and is therefore static. *How* the company does business and the technology it uses will change but these changes generally only affect development phases from Design on down. The company should not have to re-do the analysis every time the *how* changes.

Many of the techniques used by the ICMS project team came from Yourdon. His techniques are independent the development tool that is chosen for your project. Through the use of pictures, it is easy to see the flaws and holes in the analysis. They are also a very effective way to show the customer what the process looks like and to ensure the developers are correctly understanding their business. Yourdon's techniques may be dated and missing the latest "techno" buzzwords, but they do facilitate a solid understanding of *what* the system does. Analysis stands apart from the technology used to design the system. Therefore, how analysis is done does not have to change with each technological advancement. The computer industry tends to think that when technology changes, so must everything else. We need to learn to transfer over those things that are working.

What is the Analysis

Overview

Analysis is a discovery process to help the customer understand and identify their data requirements and the processes necessary to support *what* the *business* does. At the end of this process both the customer and the developers will have a mutual and succinct understanding of this *business*. The information necessary to reach this understanding can be gathered from a wide variety of different sources. As information is received it is categorized and documented. The information may relate to *what* the *business* does and *how* it is done. Both types of information are valid and need to be gathered. The key is to understand and recognize the difference between the two. Each type is very important to the project, just at different times. Analysis focuses on *what* is required to do the *business*. By separating these two concepts, the team is able to focus on *what* the true requirements are for the system and avoid getting trapped into preconceived ideas on *how* it should done. For example, NASA and the USA spent a lot of money trying to develop a pen that would work in zero gravity so the astronauts could write when in space. The business requirement is that the astronauts need to be able to write in space, but there was a preconceived idea that this had to be with a pen. When the Russians were asked how they solved this problem, they responded that they simply used a pencil.

Analysis focuses on *what* must be done and is completely separate from the technology and methods chosen to perform the *business*.

Goal

The goal of analysis is to achieve an absolute definition of *what* the *business* does, *what* data it requires to accomplish its *business*, and *what* data is generated to support the *business*. This definition includes all of the *business* and not just those portions, which may be automated. Automation is a *how*. It is void of any information about *how* the business is done, and of physical processes like monitoring, validating, verifying, or editing, and of organizational processes such as auditing.

The Information Is Out There

The information you require to develop this absolute definition is all out there. You just have to be able to find it, recognize it, and accurately piece it together.

This information can be gathered from the people who are responsible for the data (data owner), those that process any part of the data, and those that depend upon the data to perform their job. Get a copy of the current paper or electronic records for the current process. These include those supplying the data they need, worksheets used during the process, and any reports generated from the process.

One of your best sources for information will be talking to the people who are currently doing the *business*. Have them show you the steps they currently perform. This walk-through will be very helpful for many reasons. It enables you, as the Analyst, to see some of the actual data they use and at the same time, provide a good visual of *how* the job is performed. This process will also reduce the chances that steps and information will be skipped or forgotten. Look at everything they use while performing the job. Depending on the ratio their job is to the entire project will determine how much information you can expect to obtain. You will want to talk with at least one representative performing each part of the process. A cross-section of representatives is very beneficial for those tasks that have a large number of people performing it. In some cases, there will be some significant difference in how each individual performs the same job. Below are some key questions you will be trying to answer at this time. The person you talk with may not know all the answers. If they do not know, they can usually give you a lead on where this information may be found.

- What data do they need to perform their job? Who provides it to them? How do they get it?

- When found, how are issues regarding the data or the processes resolved? If you receive more than one outcome from this resolution, how is it resolved?
- What happens to the data as it goes through each step of the process? Does it change? If so, how and why?
- Why do they process the data in the manner they do?
- What tells them they need to start a particular task? How do they know they are done with a particular task?
- What kind of information do people expect from them?
- Are there any policies or procedures governing the job they do or how the job is done? If so, what are they?
- Do they always process the data in the exact same manner? Are there any exceptions to this? What causes this data to be different? How is this data processed?
- What happens with the information you generate? Who uses it? Where does it go? Why do they need it?

Again, you are interested in the entire process and not just those tasks that are already automated or planning to be automated.

Categorizing and Documenting

The information you have gathered needs to be grouped based on the impact it has on the system. Some of these categories will be defining *what* the business does while other are about *how* the business is performed. This is when the Analyst's ability to differentiate the difference between the *what* and the *how* is so critical. It sounds like it should be so simple to do, but in reality, especially when the Analyst is first acquiring the understanding of this differences, it will generate some of the more controversial issues faced by the project team.

Again, the focus here is on the process and not the tool. In many cases, more than one tool will be used during analysis. As long as the development team knows where to find the information and the symbols and terminology are interpreted the same, the particular tool used will not matter. You will find some tools are better at supporting certain documenting preferences over other tools.

Scope

One of the first documents produced from analysis is the definition of the business boundaries the project is operating in. This is the project scope. One of the simplest ways to define this is through a context diagram.

The context diagram contains the external entities that provide data to the system or require information that is generated by the system. Arrows are drawn between the entity and the system identifying the data being passed and denoting the direction of the flow. This diagram is at a very high level and should be very simplistic. Figure 1 shows a context diagram used to describe the ICMS project.

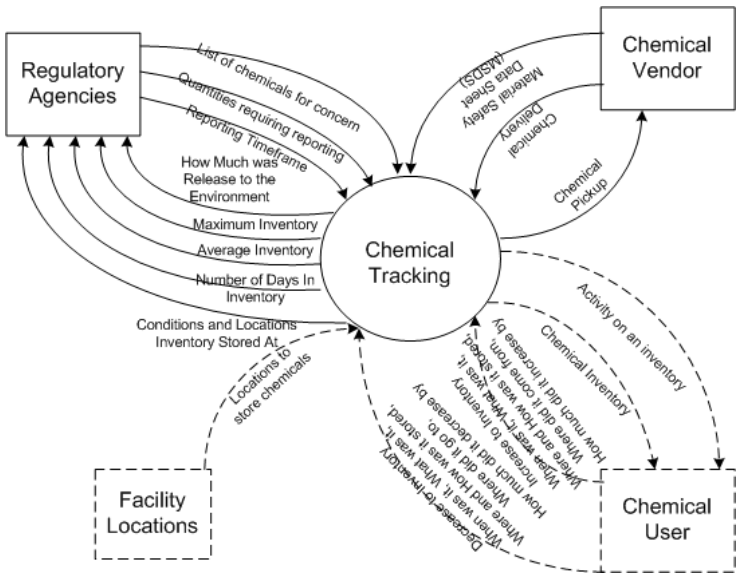


Figure 1. Context Diagram

If the context is defined correctly, both the customer and analyst should be able to agree to the following statements:

- All data required by the system is either being passed to it from an external source or will be generated internally to the system.
- All external sources using data defined by the system are shown
- The system is not concerned about any interaction that takes place between the external sources. Otherwise that interaction needs to be included in the system.
- The system has the ability to control all internal sources of data.

The context diagram is one of the first confirmations to ensure the development team and the customers have the same overall understanding of the business.

In reality, you may find some of the external sources for data may need to be included as part of the system. This can happen when the system requires information, which it does not own, and does not have access to from the source data. Figure 1 shows these by using dotted lines. It is important to acknowledge this difference so there is no confusion about which system is truly the source for this type of information. It is also an indication to re-evaluate the maintenance of this duplicate information if and when the data becomes available through the source.

In the case of ICMS, both the facility information and user or (personnel) information are external to the tracking of chemicals. But since there were multiple owners to the various sources of this data, it was necessary to include both inside the scope of chemical tracking.

The context diagram is then used as a basis for each of the subsequent models created. They must all portray the same statement of *what* is required by and generated from the *business*. Cross-checks should be done throughout the entire analysis process. The development of these subsequent models will occasionally identify a change required to the context diagram, but generally the context remains fairly stable once it has been agreed upon with the customer. This scope is defining what is being evaluated by this project. Items not defined in this scope that are being given consideration are a good indicator for "scope creep", which in turn, will postpone or prevent the completion of the project.

System Requirements

The data requirements and process requirements together make up the system requirements.

Data Dictionary

As you collect information the system is interested in you start recording what this information is and assign it a definition in the data dictionary. Most of these terms will eventually end up somewhere on the ERD, but the dictionary also contains the precise definitions. For non-OO modeling, this would also be a place to record the business rules around the data. This information is part of the analysis but most modeling techniques do not provide a place to record it. Typically this type of information will eventually be enforced through a constraint or trigger, but as in the case of Designer documenting this information is not available until the Design.

Data Requirements

This is the logical model of the data required for the system. The most common method for defining and documenting the data requirements is through an Entity Relationship Diagram (ERD). Generally you begin by defining entities that are of interest to the system. The inputs and outputs from the context diagram are a good place to start but probably will not be all of them. You can then start defining how these entities relate to each other. New entities will appear to resolve the many-to-many relationships that exist. From there, the attributes pertaining to the entities can be assigned and possible primary keys identified. The entire time you are doing this you should be comparing this with the context diagram. Verify that all information required by the system is on the ERD and that all other information is generated by the system.

Figure 2 shows a portion of the ERD created for the ICMS application.

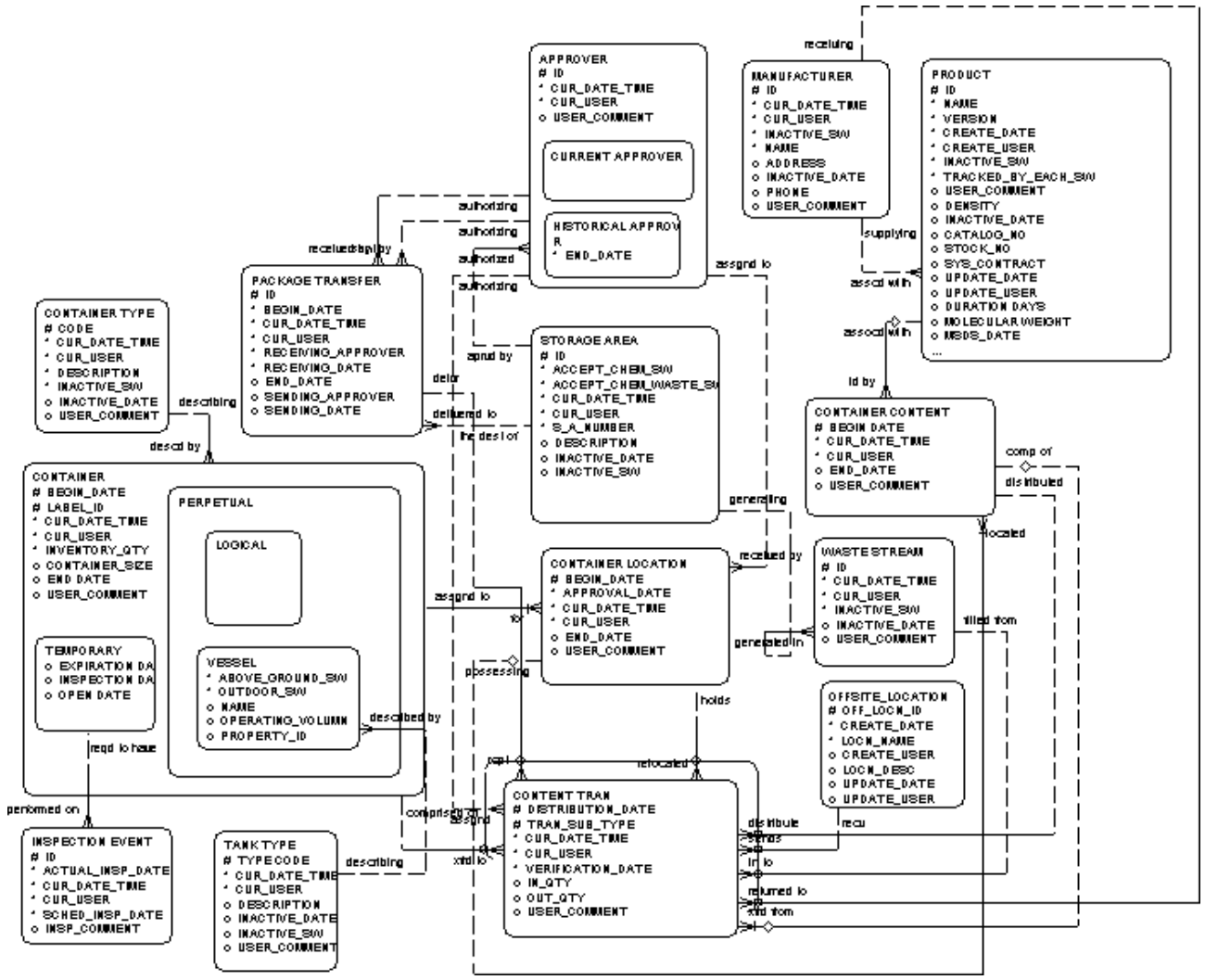


Figure 2. Entity Relationship Diagram

Process Requirements

These are the logical models and represent the data and processes required to complete the defined scope. You need to start with the context diagram and break the system down by levels of detail. Again these should be compared with the context diagram and the ERD. All data required by the system should be seen as an input into one or more of these processes and all information generated from the system should be seen as an output from one and only one of the processes. Only the minimum data required for the process should be included. This is also called starving the process. If any of the processes or data are removed from the system, the system would not be able to perform the required work.

These processes are also sometimes called data transformers. Each process takes in data, transforms it, and provides new data out. Physical processes such as monitoring, editing, or validating are not included here. Neither are organizational processes such as auditing. They do not transform data but tend to support accuracy and accountability.

Figure 3 shows one of the higher level flows describing how a chemical moves through the facility.

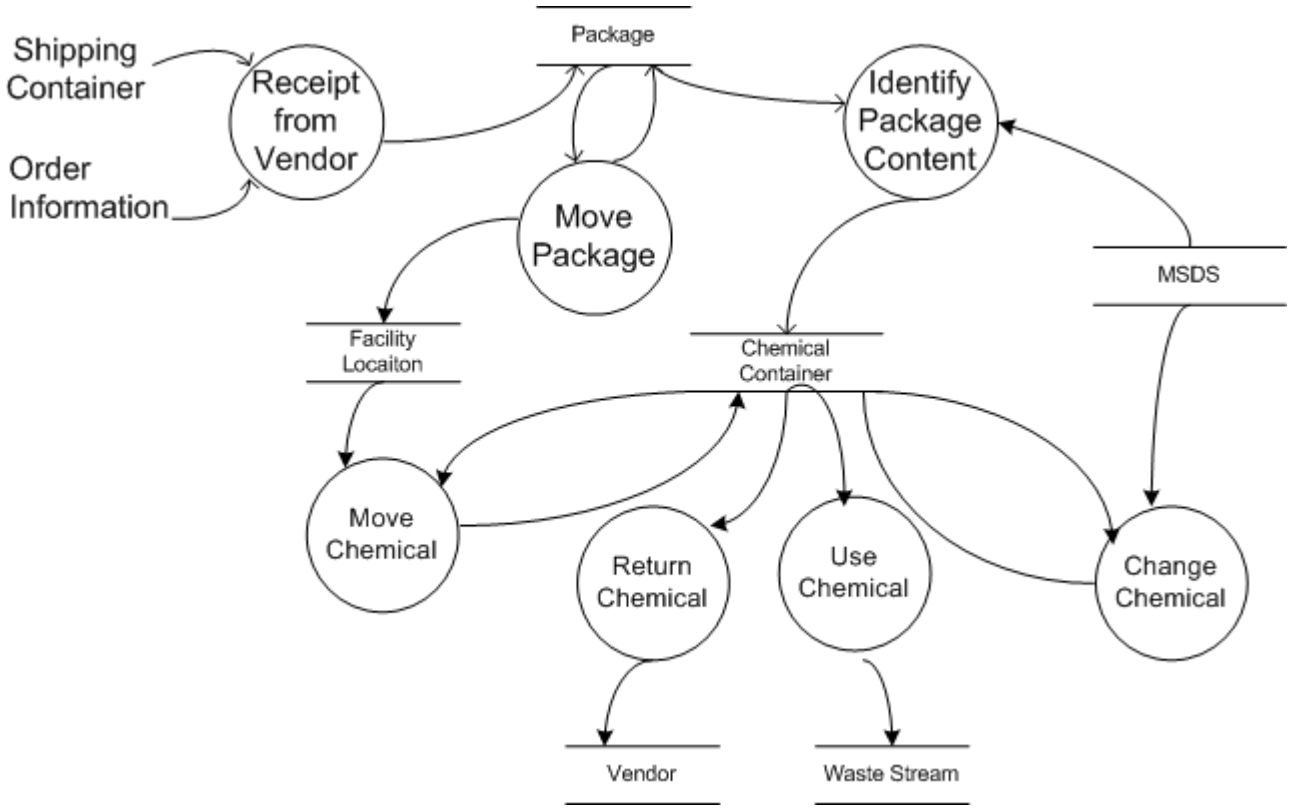


Figure 3. Process Flow Diagram

Constraints

Constraints are rules placed on the system through a written document. They are still rules the system must follow but they are defining why the business does the requirements. They are always published and should point back to the constraining document. They are always external to the business requirements and generally change more often. These should almost always be implemented as data driven.

Some of the constraints identified for the ICMS project are:

- 40 CFR 355 – EPCRA 311 – Chemicals stored above the threshold limit must be report within 90 days
- 40 CFR 370 – EPCRA 312 – Maximum, Average, and storage conditions for chemicals exceeding the maximum threshold limit any time during the calendar year
- 40 CFR 313 – EPCRA 313 – Quantity of toxic chemicals used during the year and how much was released to the environment
- IDAPA 58.01.01.322.11 and Environmental Oversight and Monitor Agreement between US-DOE and State of ID – Reporting of all air emissions which exceed the permitted limits.

Objectives

Objectives are items the customer would like the system to do. These are going to impact *how* the system will work. In actuality, these are all negotiable rules. They are not really addressed in analysis but will come up during this phase and need to be documented. A dollar amount should be able to be associated with each and the customer should be able to rank them in order of importance. Typically, requirements relating to a specific technology are really an objective.

Some of the objectives for ICMS were:

- Identify what is in a chemical container based on a barcode label that has been attached
- Electronic notification to custodian with pending chemical actions.
- Chemical requests can be initiated available via the Web.

- Current chemical inventory available via the Web.
- An indicator to the custodian that they are working with an EPCRA 313 toxic chemical.

Problems/Needs

These are issues people may have with the current way of doing the *business*. Generally efficiency concerns, system availability requirements, and response time issues are documented here. Again these are Design issues and will be considered in that phase. But since they typically come up during the Analysis, they should be documented and passed on.

Some of the Problems and Needs identified with ICMS were:

- Custodians responsible for the chemical of tracking do not receive the paper work signifying the receipts and issues in any particular order. Paperwork recording receipts may lag significantly behind the paperwork recording the issues out.
- Warehouse personnel who receive the chemicals are not trained to compare and recognize discrepancies between the chemical ordered and the chemical received.
- Chemicals may simply appear in a custodians cabinet and there is identification of where it came from
- Custodians tasked with tracking chemicals are seldom notified when a container of shock-sensitive chemical is opened. Shock-sensitive chemicals must be evaluated periodically after opening for peroxides.

Common Pitfalls

The majority of problems encountered in systems development can generally be mapped back to a flaw in the analysis. Unless the flaw is fixed, it will continue to cause an increasing number of problems. Modifications, manual or automated, that are set up to compensate for the flaw will become increasingly more complex with each change or revision, until eventually the system will no longer be able to function with the flaw. Until then, the company will be spending both time and money trying to maintain and use the system.

The problems and outcomes from weak analysis are encountered and fought on a daily basis in our industry. In my opinion, the majority of all development dollars being spent today are not providing the benefit they should and could provide to the company because of weak analysis. Many people in the computer industry will tell you that analysis is important, but their actions do not substantiate this acceptance. I attribute this to the following observations:

1. Analysis provides a long-term return on investment; businesses operate in a short-term window.
2. Customer doesn't have time to explain what they do. That is why they are having someone else create the system.
3. Customer wants to see screens and reports, not pictures.
4. There is a chance the flaws will not be noticed until a change is needed. Then it might be someone else's problem.
5. Can provide job security by keeping the customer dependent upon the developers in order to accomplish his job.
6. Developers do not always get the opportunity to assimilate from lessons learned. This can be attributed to shops that have independent groups responsible for each phase of the project; or where a developer is frequently moved between projects.
7. Systems usually have unrealistic deadlines set and eliminating the Analysis will enable developers to get a jump start allowing them a cushion to address problems they find.
8. Programmers want to get coding.
9. Analysis is not required to write a system.
10. People think that if they have access to development software it must mean they can develop systems.
11. There is a widespread misconception that Analysis and Design is the same.
12. Design is where the technology is used. People want to use the latest and greatest tools and it is generally seen as more fun.

How is the Design

Overview

Design takes *what* the business does and provides an interface for doing this business. If defines *how* the business is performed. This is building the new physical model. It compromises the logical model for the sake of understanding,

efficiencies, and ease of use. It takes into consideration how to support the objectives and handle the problems and needs identified. But it does not alter any of the definitions supporting *what* the business does. And any business that was not part of the scope during analysis should not be included in at this time.

Design is where I feel methodologies; such as prototyping, can be brought in and used. This is the part of the system that is impacted by the changes in business practices and technology. The *how* is what needs to be able to adapt with time. Systems should be designed so that it is the data that drives *how* the system operates and not from within the code running the system. This will provide the maximum flexibility and growth for the customer.

Goal

The goal of design is to achieve the most efficient and simplest way for performing the business, while keeping it feasible and cost effective to accomplish. This is the *how*. It brings in the physical processes needed for accuracy. It includes the organizational processes needed for accountability. It takes into account technology available to reduce the time to perform the business. It also uses industry best practices to facilitate in the development of the system and reduce the effort needed to maintain and expand the system.

Objectives

Objectives play a major role in the design. Which of these are actually included is completely in the hands of the customer. Given enough time and money anything is possible, they have to decide which are worth pursuing. These are options that decide the balance between the three factors of a project; time, money, and quality. Realize the customer owns this decision. Some of the items listed as an objective may be looked at as a requirement from the customer's standpoint. But simply stated, this means they understand the resources to implement and want the objective implemented or that the effort to implement is worth it regardless of the time and cost.

Problems and Needs

Problems and needs are taken into consideration and should be done if feasible. They usually have to do with the efficiency, ease of use, and the accuracy of the system. These issues could have a major impact when choosing from various design options available. One choice may make it easy or at least possible to resolve some of the items where another choice could make it impossible to resolve.

Constraints and Other Data Driven Information

Let the data drive the system, not the code. This means that *how* the system operates is dependent upon what the data is and not on the actual value of the data. This will handle a large majority of the changes a customer typically requests. I sometimes refer to this as "controlled" data. It defines *how* the business is accomplished. The customer already owns and is responsible for the information the system facilitates, they should also have the ability to change and manage how it operates without needing the "computer" staff.

No longer can the application be blamed for inept or insufficient decisions on how the business is done. The person who controls the "controlled" data owns these issues. The more control over this data, the more tightly the customer can enforce how the business is done. The greater this access, the less control they will have.

Examples of data used to drive the ICMS system are:

- Factors used to convert from one unit of measure to another, within the same unit type (mass, volumne)
- Factors used to convert from one unit of measure type to another
- Waste streams available for the user to define how and where a chemical was used. The options available will drive how much detail is available for retrieving the information back out. Many specific ones will allow for greater differentiation when evaluating and reporting the data but will be more confusing for the user and it will increase the time required to enter the transaction. Fewer more generic ones will be easier to select from but will only be able to provide the information back out at this generic level. But this is how the customer controls their system.

Domains are also a way to manage the allowable values, but should only be used for completely static data, such as "gender" or "true/false" values. Values available for all other lists should be implemented through the data and should be managed by the customer. Changes in the business climate and technology will frequently provide new options for many of the lists that are considered "static" in today's current environment. This provides a means for the system to easily adapt to most "environmental" changes.

Data Definitions Do Not Change

All of the data defined and brought forward from Analysis must exist somewhere in the Design. The Design will most likely require additional data items be added, but it should not change the definition of the data that is required to support *what* the *business* does. The temptation to do this could be indicating or hinting at a flaw in the analysis. This possibility should be investigated so that errors are caught as early as possible and prevent any further escalation.

Changing the meaning of data is also quite common during the maintenance of or the enhancement to the system. This problem is inevitable if these definitions are not adequately documented. People and memories always change with time. Sometimes merely changing the definition may be seen as less of an impact to the "current" system or thought to be less confusing. But this should never be done without first evaluating the impact this change has on the analysis. When only considering *how* the company performs their business many of the changes will appear harmless, when in reality the change could hinder or prevent accomplishing *what* the business does.

Below is an example how this issue affected ICMS during one of our enhancement projects and then again by the customer:

Original Definition

The Analysis of chemical tracking required knowing if the person performing the chemical transactions was authorized to do so. The Design chose to drive this requirement using data values entered into the database. The data defines where each person is authorized to perform chemical transactions. The system knows who the person is based the user's logon id and password. This relationship is based on a person's ICMS User ID.

As stated earlier, ICMS does not really own the information about people but since this information was not accessible from the real owner it had to be included as part of this system. The INEEL has a company policy that states certain requirements must be met before someone is given access to any computer systems inside our firewall. But, whether a person meets these requirements or not, did not have to be included. There is an organization that ensures these requirements are met and then will assign the person with their "preferred" computer id. So, the ICMS has a manual process to obtain this id and then enter it into the USER_ID field on the PERSONNEL table. USER_ID is this table's primary key.

Enhancement Modification

Due to an enhancement, ICMS was required to know any person requesting chemicals. This meant everyone with access to the Intranet. The enhancement was Designed to generate and send an e-mail message notifying to the requester on it's status as it progresses through the process.

This meant ICMS needed to have an entry for everyone with access to the INEEL's Intranet. And by Design it was required to know an e-mail address for each person. Since the "preferred" computer id is also generally used as the person's e-mail id and in the spirit of keeping things simple, the definition of USER_ID on the PERSONNEL table was changed to represent the "mail id" of a person on the host domain @inel.gov. But even this definition did not meet the full requirement because not everyone receives their mail through the @inel.gov host domain.

But the major impairments occurred because the impact of this change was not evaluated against *what* the system required. The existing part of the system still needs to identify who is logged on so authorizations can be verified. This field is a critical factor driving how the system will operate. The new enhancement needs to know where to send the statuses for a request. And again, this field is the driving force. As long as the values for each of the definitions are the same, there is not a problem. But what happens when they are not? The system still requires both pieces of information to operate, but there is only one place to record the two different values. This means the system will not be able to perform one of the two functions required of it.

Customer Impact

Definitions can also be inadvertently changed by the customer with data driven items. This is why it is important that the customer clearly understands how these particular items impact the system's operation. Information that is not consistent with the definition could prevent the system from being able to operate as needed. It will also add to the complexity of future enhancements.

Again with ICMS, the definition of USER_ID was almost changed through this method. One of the users did not want the mail coming to her particular e-mail address. She wanted to route her mail to a generic e-mail address so that everyone in her group could access the messages. This would have been very simple had a place been created to record each individual's e-mail address. But since the place to record this type of information is shared with the definition of a single person, it was going to cause problems and a different resolution had to be developed.

One of the primary problems it would have caused was in the future development. ICMS does not own the people information. Entering values that are not representative of a person would make it very difficult or prevent being able to elevate this duplication and begin using the source information.

Summary

Sound Analysis and Design practices are crucial to give the application the ability to adapt in today's ever changing business and technology climate. Below is a list of statements that need to be followed to receive maximum benefit from your "computer" dollar.

- Your company's rate of return from their technology dollar will depend upon the quality of the analysis performed.
- The customer and users are an integral part in the development process.
- Analysis has to be done. You can do it now or later. And it is quicker and cheaper to just do it first.
- *What* the business does is not the same as *how* it is done. Keep them separated.
- *What* a business does is fairly static. *How* it goes about doing this business is what is dynamic.
- Analysis is independent of the tool you choose for recording the findings.
- Analysis is intended to describe *what* the business must do and the data required to do it.
- Requirements are created by piecing together information. Generally this information will come from more than one source.
- Major deliverables produced during analysis:
    1. Project scope
    2. Data Requirements
    3. Process Requirements
    4. Constraints
    5. Objectives
    6. Problems and Needs
    7. Data Dictionary
- The project scope will help a project stay on task by avoiding possible scope creep.
- The project scope defines what is internal to the project and what is external.
- Data requirements are a Logical model of the data needed by the system.
- Process requirements model the processes and minimum data needed for the system to work.
- The data dictionary is a good place to record the business rules around the data if the tool you are using does not provide one.
- Constraints are external rules imposed on the business and should be implemented through the data.
- Constraints are always in writing. Rules defined in constraints are defined through the document and not the specifics listed in the document.
- Objectives are always negotiable based on time and money and the customer should be able to prioritize them in order of importance.
- Actions by most in the computer industry show that they feel Analysis as an optional phase.
- Compromises to the theoretically optimum solution are imposed at Design time to provide an interface that the people involved with the system can understand.
- Definitions for required data are defined during Analysis. Design should not compromise this meaning.
- The decisions made during Design should support the information generated through the Analysis.
- The data should drive the system and the customer should be the one with the ability to manage these factors.

Definitions

- *Business* – The breadth of this term is defined by the project scope.

- *Data* – By itself it does not have meaning. They are characters or numbers. A method for recording.
- *How* – Defines the technology and processes used to complete a process
- *Information* – Data that has been evaluated and used to enhance knowledge
- What – Defines the minimum steps required to complete a process.

About the Author

Judy Bramlette is a Principle Technical Specialist at the Idaho National Engineering and Environmental Laboratory. Judy has 19 years experience in Information Technology, which the last seven have been using Oracle/Case Designer. She is involved in all aspects of the development process. She was one of the original developers on the ICMS project and continues with its the maintenance and enhancement efforts. Judy has a B.S. in Business from Montana State University.

Copyright Notice